



Listado con Statement #2

Transcripción

[00:00] Bueno, volvemos aquí al `ProductoController` y para tomar el resultado del statement que nosotros recién ejecutamos, nosotros tenemos que ejecutar otro comando. El comando es del statement, en el propio `statement.getResultSet`. Este método nos devuelve un objeto del tipo `ResultSet` también.

[00:29] Bueno, ahora tenemos un objeto más y otro desafío. ¿Cómo vamos a leerlo para tomar el listado de los dos productos para devolver para la pantalla de la aplicación? Bueno, un `resultSet` es un listado de resultado y siempre que lo vamos a leer tenemos que saber cuál es el próximo elemento desde la fila actual, o sea, mientras haya una fila en este `resultSet` nosotros podemos ir leyendo el resultado.

[00:56] Cuando lleguemos al último ítem de este `resultSet` el loop se termina, o sea, si estamos hablando de un loop nosotros podemos iterar este objeto. Entonces el resultado nos provee una forma de ir hasta el próximo elemento del resultado, hasta que lleguemos al final. El recurso que nos que nos provee `resultSet` es un método llamado acá `resultSet.next()`.

[01:26] Parecido con el iterator. `.next`. Y eso lo podemos agregar dentro de un loop de `while`, `while (resultSet.next)`. Nosotros vamos leyendo su contenido para poder agregar a un listado de resultado y devolverlo para el front de la aplicación. Pero tenemos un listado de algo para devolver.

[01:50] Voy a borrar acá este result que no vamos a estar utilizando más. Y tenemos un listado para devolver. ¿Pero un listado de qué? Bueno, en este caso vamos a hacer lo siguiente, yo voy a extraer eso acá para una variable. Ahí tenemos un resultado. Y este list, que no es un ArrayList, va a ser un list, lo voy a declarar acá arriba del while.

[02:22] Hacemos acá un list. No va a ser de objeto, va a ser un link de un map, ya que no tenemos nada para representar nuestro producto, vamos a estar utilizando un map ahora del tipo String, String y devolvemos un listado de este tipo para nuestro front para que haga el manejo de los datos.

[02:48] Bueno, y ahora acá, volviendo para resultSet porque tenemos que agregar este resultSet a un map para que podamos agregar al listado. Para cada registro de resultSet, para cada fila de ese resultSet, nosotros podemos tomar la información de las columnas de la query, que son, ID, nombre, descripción y cantidad.

[03:07] Bueno, para los tipos ID y cantidad, que son numéricos, nosotros tenemos un método acá resultSet.getInt. O sea, este método getInt es el método que nos devuelve la parte numérica de la query, que es el ID y la cantidad. Y acá tenemos dos formas de llamar, que es con un entero, que es el índice de la columna, empezando por uno. O sea, el ID es el índice 1, el nombre índice 2, descripción 3 cantidad 4.

[03:44] O podemos estar utilizando el propio nombre de la columna que declaramos en la query. Sería entonces resultSet.getInt("ID"); y ahí estamos con uno de los resultados ya. Esto lo vamos a asignar a un map, entonces voy a declarar acá un map del tipo String, String que es una fila = new HashMap. Ahí está.

[04:11] Y bueno, esto acá lo podría sacar. Ahora sí. Y bueno, esta fila, nosotros vamos a estar haciendo un fila.put, la clave va a ser "ID" y el valor va a ser este resultSet.getInt("ID"). Siguiendo acá. ¿Ahora de qué se queja? Se queja porque

estoy agregando un entero para un campo string, entonces hago acá un `String.valueOf`. Se borró todo ahora. A ver otra vez.

[04:50] `String.valueOf`. Ahora sí, ahí está. Hacemos acá `fila.put("ID")` y vamos a replicar lo mismo para las otras columnas, entonces hacemos `fila.put("NOMBRE")`, y para tomar el nombre, nosotros tenemos si para valores numéricos tenemos `getInt`, para valores que son del tipo string, tenemos el `resultSet.getString`.

[05:26] Ahí está el `getString`, y podemos poner también el nombre, el propio nombre de la columna. Para los otros campos, igual. Entonces va a ser lo siguiente acá. Voy a copiar este comando, lo pegó acá y cambio el nombre para descripción. Copio acá DESCRIPCIÓN y en el nombre acá pego acá DESCRIPCIÓN.

[05:49] Y por último, lo de cantidad, que es igual que el ID, que es un `Int`. Voy a copiar la fila del `Int` y cambio acá, lo que es allí para cantidad. Y allí acá cantidad también. Por fin hago un `resultado.add(fila)`; y ahí está, para cada registro del `resultSet` nosotros estamos transfiriendo la información para un map, un objeto del tipo map, y lo agregamos para el listado que es el resultado.

[06:22] Y al final cerramos la conexión y devolvemos la información. Bueno, ahí terminamos la lógica. Bueno, voy a agrandar acá la pantalla así la ven bien. Como estamos devolviendo un resultado que es un tipo, un list de map Strings Stream, acá en la declaración del método voy a sacar a esta interrogación acá y voy a agregar el `maple String`, `String` también, así dejamos declarado que el método devuelve un listado de este tipo.

[06:51] Y ahora vamos para el método de cargar tabla, para cuidar de eso para nosotros. Bueno, acá tenemos ya los productos que están en un try catch, eso tenemos que arreglar un poquito. Esta información de acá voy a dejarla acá, dentro del try catch, también. Y tenemos de este método acá en to do, que tenemos que finalizar ahora para poder enviar los valores para la pantalla.

[07:20] Vamos a hacerlo entonces y ahí ejecutamos la aplicación para ver cómo queda el resultado. Bueno, acá en el to do voy a hacer lo siguiente, voy a sacar el comentario. Y acá tenemos un `addRow` que pone un objeto de ID, nombre, descripción y la cantidad que me olvidé acá de poner la cantidad, la vamos a agregar.

[07:39] Entonces, como tenemos varios productos, tenemos ya el producto es `.forEach`, o sea, para cada producto vamos a estar haciendo `producto.get`. Y ahí ponemos la clave, ID. Y después tenemos `producto.get("NOMBRE")`. Después `producto.get("DESCRIPCION")`: Y por fin, `producto.get("CANTIDAD")`. Ahí está.

[08:25] Tenemos las cuatro columnas y para arreglar una cosa en donde llamamos el cargar tabla y me olvidé también, tenemos que a ver, ¿dónde está? Acá ya tenemos un label de cantidad, pero eso es del campo del formulario en donde cargamos la tabla, tenemos que poner el identificador del producto, el nombre del producto y acá el `modelo.addColumn("Cantidad")`; ahí está.

[09:04] Ahora guardamos todo, vamos a ejecutar la aplicación y ver cómo quedó el resultado. Voy a dar un play acá en la partecita de acá de arriba. Vamos a ver qué dice en la consola, si nos muestra algo. Bueno, hoy estaba levantando la aplicación. Y acá tenemos nuestra aplicación con el listado de productos de la base de datos cargados acá en la pantalla, directamente.

[09:26] Listo, y ahora tenemos aquí en la aplicación que está cargada ya en la pantalla, el listado con los productos de la base de datos también ahí presentados en nuestra pantalla de la aplicación Java, la mesa y el celular. El objetivo es seguir complementando las funcionalidades de la aplicación con las demás operaciones.

[09:46] El delete, el insert y el update, y otras operaciones que vamos aprendiendo en el camino también. Entonces, por aquí quedamos y nos vemos en la próxima clase.

